

EECS, KTH

DD1319 prgcl hösten 2025 (???)

200 Pokemon

P-uppgiften ska göras individuellt. Läs EECS:s hederskodex innan du börjar!

Varudeklaration: Klasser, datastrukturer, slumpstal, filhantering, interaktiv GUI,

Jag ska skapa ett pokemon spel. Syftet med spelet är att skapa en simplifierad pokemon combat simulation. Det finns inget "vinnande" i spelet utan spelaren förbestämmer själv när de vill avsluta. Spelet kommer fokusera på två huvuddelar: combat och "pokemon hantering". Datan för pokemonen ska läsas in från en .csv fil. Filens format kommer vara i form av:

name, level, attack, health, defense, can evolve, **#kommer eventuellt ha mer data**

pikachu, 1, 10, 10, 10, True

mewtwo, 1, 15, 10, 5, False

Combat:

Spelaren kommer ha ett lag på max 6 pokemon. Det ska finnas en liten rektangulär spelplan som spelaren kan gå runt på. När spelaren går runt finns det en slumpad chans att de stötar på en "encounter". Alla slumpade saker är just nu obestämd och behöver testas vid körning för att se vad som funkar bäst.

En encounter är en turn-based combat system där spelaren skickar en av sina pokemon för att slåss mot en annan. Spelaren kommer ha fyra val:

- Attackera:
 - Skada motstående pokemon
- Byta pokemon:
 - Byta till en annan pokemon spelare har på sitt lag
- Fly från encounter
 - Avsluta encounter utan combat och gå tillbaka till spelplan
- Items
 - Healing item för att återställa sin pokemons hp
 - Pokeboll för att fånga motstående pokemon
 - Alla items kommer finnas i begränsad mängd förnyas vid varje ny encounter

Om alla spelarens eller motstående pokemons hp når 0 avslutas encounter:n. Spelarens val av action kommer vara i form av en meny:

0 : attackera

1 : byta

2 : fly

3 : items

Motståndar pokemonen samt dens level kommer genereras slumpmässigt i en range som är nära spelarens pokemons medel-level. Attackerna som används av motståndar pokemonen kommer också vara slumpad.

Pokemon hantering:

Pokemonen ska vara skriven i klasser där "Pokemon". Huvudattributen som klassen ska innehålla är:

- health points
- attack skada
- speed (bestämmer vilken pokemon som attackerar först)
- moveset (någon sorts "built-in collection type" som innehåller max 4 olika attacker)
- 1-2 types
- level

Kommer även ha mer men det här är de viktigaste.

När en pokemon vinner i en fight kommer den få exp och eventuellt levla upp och evolva när den når en specifik level. Man kommer också kunna fånga pokemon för sitt lag. Man kommer också ha möjlighet att "slänga" pokemon ifall man vill fånga fler och man redan har 6 st.

Spelet ska se ut på liknande sätt:

#Encounter

You've encounter a wild {opposing pokemon}

Go {your pokemon}!

What would you like to do?

[0] : Attack

[1] : Swap

[2] : Items

[3] : Flee

: 0 #Användar input

Which attack?

[0] : Light Attack

[1] : Heavy Attack

[x] : Back

: 0

{your pokemon} used {attack name}

{opposing pokemon} took {damage} amount of damage. {hp}/{max hp}

{opposing pokemon} used {attack name}

{your pokemon} took {damage} amount of damage. {hp}/{max hp}

What would you like to do?

...

: 1 #Användar input

Swap to?

[0] : Pokemon 1

[1] : Pokemon 2

[2] : Empty

[3] : Empty

[4] : Empty

[x] : Back

{opposing pokemon} took {damage} amount of damage. 0/{max hp}

{opposing pokemon} fainted!

{your pokemon} gained {exp} amount of experience! Level: {level}, Exp: {exp}/{exp needed to level up}

#Spelplan

```

*****
*-----*
*----P----*
*-----*
*-----*
*****

```

[0] : Upp
 [1] : Down
 [2] : Right
 [3] : Left
 [4] : My Pokemon
 : 0

```

*****
*-----P-----*
*-----*
*-----*
*-----*
*****

```

Gå:

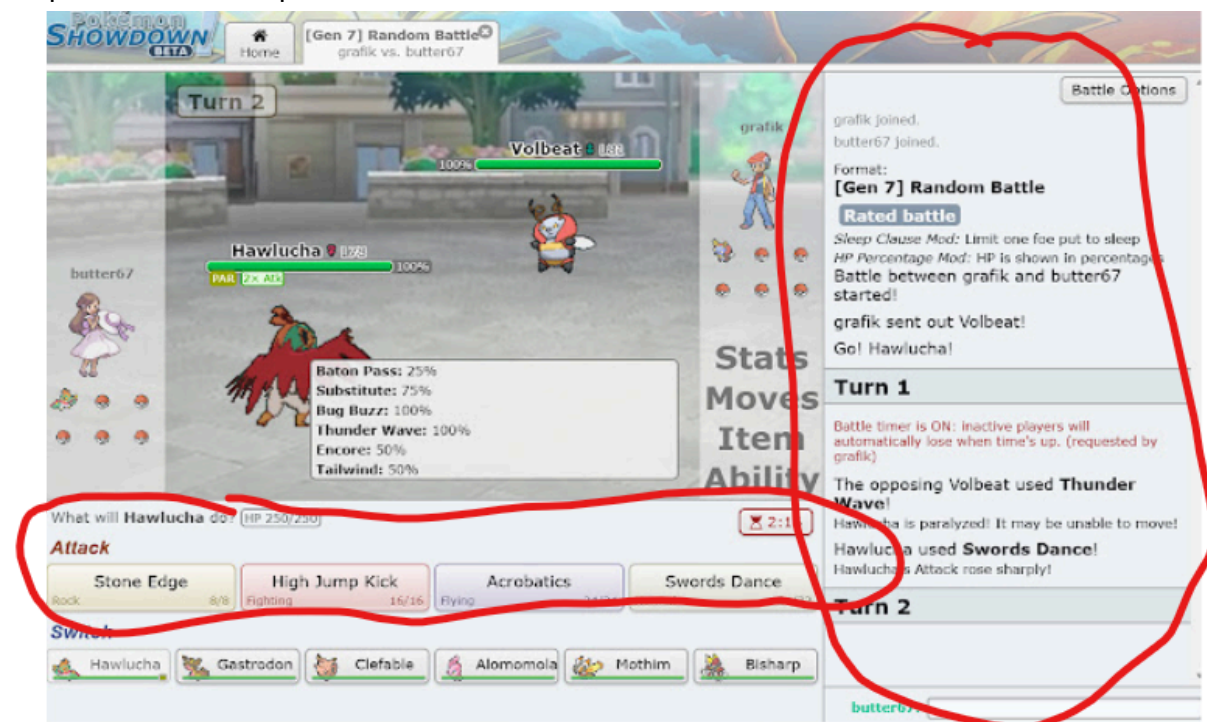
...

Med tkinter kan man tillämpa keyboard controls så användaren kan använda pilar

Implementering av användargränssnitt kommer vara i form av att användaren kan använda knappar på tangentbordet för att gå runt (ex. pilar). Och knappar på skärmen istället för att skriva in 0, 1, 2 eller 3

Resterande kommer vara i text format liknande till terminalen, det kommer INTE innehålla visualisering på pokemonen såsom sprites.

Inspirationen är från pokemon showdown:



Extra:

C | Hantera felkontroll, ex. kunna ta fel inputs utan att krascha

B | Låter användaren spara dens nuvarande pokemon lag i en ny fil för att sparas. Filen ska även formateras rätt så den kan användas igen

A | Grafiskt användargränssnitt med tkinter eller liknande